Al Imam Muhammad ibn Saud Islamic University

College of Computer and Information System

Computer Science Department

2nd Semester 1444 H – 2023 G

CS 346 – Web Development

Project

قاوم

BY

STUDENTS	ID
NJOUD ALOTAIBI	
TEEF ALOSHAN	
REEM AL-OTHMAN	
LOJAIN ALZAIDE	
ALJOHARA ALDAWOOD	

[12/2/2023]

Qaweem Website

In this project, it is the integrated solution for providing care and psychological well-being services remotely, through sessions, programs, and a private space for expression. The site targets everyone who has a need for psychological assistance and anyone who has confusion or confusion about anything.

In the event of a need for a session, the user can choose the specialist by viewing the profile, booking an appointment and time for the consultation, and the specialist completes to reach a solution that helps the person

There are also different programs that the user can join to improve a problem he suffers from

Through the expression space, the user has two options for expression in the writing section or in the drawing section, where they can choose whether they want it in a private way that does not reach anyone or on general so that a specialist can see it and be able to help, in order to allow the user to be in control and have the ability to make his/her own decisions. To feel comfortable expressing it.

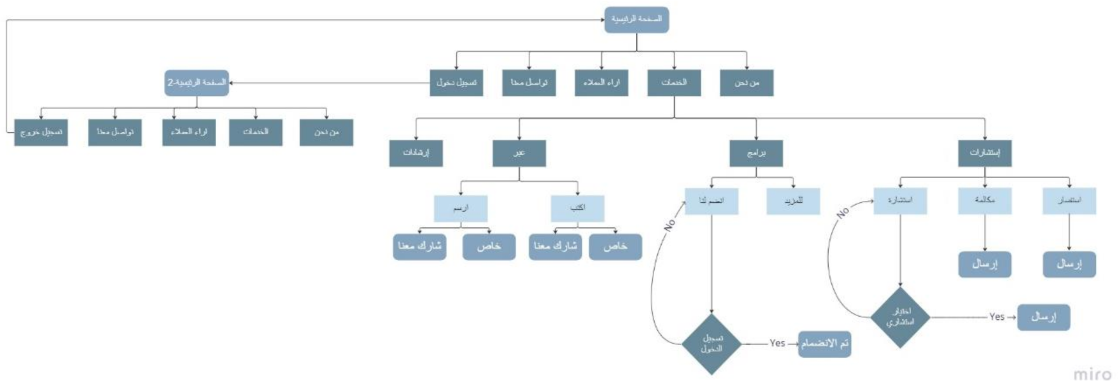
There is a space for instructions that enable the user to view and read them.

always (Qaweem) is your first step to change.

On this project our classes consist of and the main programs:

- Html
- Css
- JavaScript
- Nodejs
- Mongodb

1. Flow Chart



Refences :

1. https://miro.com/app/dashboard/

2. Look & Feel

Logo

The logo is Resist in Arabic “Qaweem”. It’s designed with a zigzag line that illustrates the failures, successes, and stages that every person goes through.

Name

The name (**Qaweem**) means to resist in Arabic, which we chosen because the basis to continue life is through resistance and not stopping attempts. Here, resist “Qaweem” helps you to continue life and teaches you on learning the skill of resisting. Qaweem is your first step to change.

Colors

The shades of blue and white were adopted for the entire design, in addition to that all the images were the same colors and reflected a good feeling for the user to give comfort and relaxing feelings, such as (sky, clouds, mountains, and ocean) our background images are nature. Nature generates many positive emotions, such as calmness, joy, and creativity and can facilitate concentration. Which allows the user to express their feelings freely.

Refences :

1. https://seekcoding.blogspot.com/2021/03/how-to-create-login-registration-form.html
2. https://www.free-css.com/free-css-templates/page274/agency-perfect

3. Dynamic Components

All the classes that are in our project:

- draw2.ejs
- write.ejs
- consulit.ejs
- doit.ejs
- Express.ejs
- Home.ejs
- Home2.ejs
- Program.ejs
- Signup.ejs
- Pro1.ejs
- Pro2.ejs
- Pro3.ejs
- Login.ejs
- app.js
- db.js
- token.js
- authMiddleware.js
- User.js
- common.js
- Verify.js
- Email.js

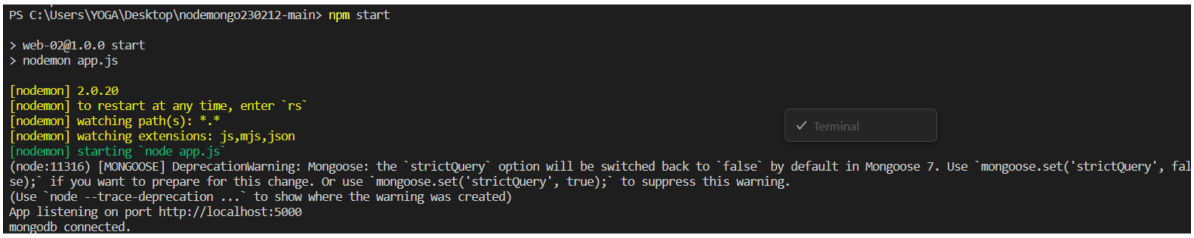
4. Business Logic

Node Js: The version that was downloaded on our device is v16.18.0, after make sure that it is downloaded we convert all our html file to ejs. Then we change a few things within the classes. Then we applied the following command Node app.js to run the program, after that a localhost://5000 has appeared, and then copy and past it on to a tap in google or any browser for the website Qaweem to display.

The Database structure used in this project is as follows: firstly, we created a Database using mongoose tool provided by the MongoDb, after creating a database, a Schema has been created in order to store all the data that will be provided by the user. Collections have been created using Mongoshell tool and linked to their schemas. Finally the queries will be collected when the user provides their data for both the signup and the login forms.

We also have used mongodb Compass and this will display all my collections with the data, which in our project we have the database folder named Qaweem and in it there are two collections one for signup and the other is for sign in. if enter the first collection we will see all the sign up data of the users information such as first name (fname), last name (lname), email, and password (pass). On collection2 we will see the sign in data of the users information such as email and password (pass). Mongodb Compass also gives us the localhost of the mongodb and we used that on our code to make sure that all data is collected and store on that specific database. The sign up and login form works when the user presses the button the schema insert the following information: Fname, Lname, email and password, to the database that we create. Now when the user is registered to the web, the user can make sign up. The database(mongoose) well check if the user exist it in the database or no. When Qaweem is pressed and then press on the LoginSignup for a file to show named users and through that shows the register information of each user registered.

We installed Mongodb version 6 but we have discovered some problems that we have faced due to the version being low, so we installed an older version which is mongodb version 4.4. after that we connected to the mongodb just like figure1.0, after entering the command “start npm”.



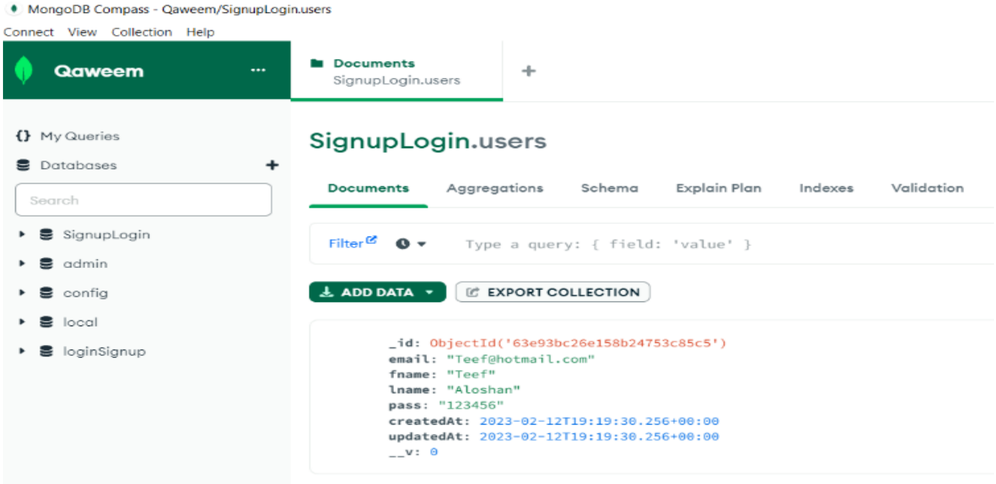
```
PS C:\Users\YOGA\Desktop\nodemongo230212-main> npm start

> web-02@1.0.0 start
> nodemon app.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
(node:11316) [MONGOOSE] DeprecationWarning: Mongoose: the `strictQuery` option will be switched back to `false` by default in Mongoose 7. Use `mongoose.set('strictQuery', fal
se);` if you want to prepare for this change. Or use `mongoose.set('strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
App listening on port http://localhost:5000
mongodb connected.
```

Figure 1.0 (Mongodb is connected)

Mongoshell is another way to connect and see the database through it. The commands that we used on Mongoshell to show database:

1. Show dbs
    - this shows all the database
    - then one of the database that appear is SignupLogin
2. use SignupLogin
    - now we can see what is in this specific database
3. show collections
    - this will show us all the collections which are
        - collection1
        - collection2

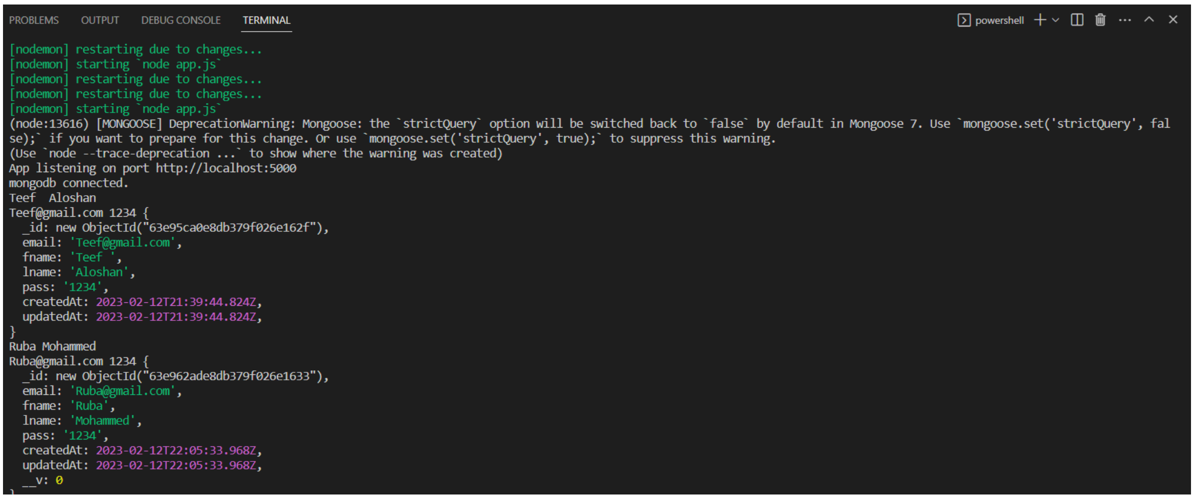
But we used the MongoDB Compass to display the database as shown in figure 2.0.



```
_id: ObjectId('63e93bc26e158b24753c85c5')
email: "Teef@hotmail.com"
fname: "Teef"
lname: "Aloshan"
pass: "123456"
createdAt: 2023-02-12T19:19:30.256+00:00
updatedAt: 2023-02-12T19:19:30.256+00:00
__v: 0
```

Figure2.0 (the register info on MongoDBCompass)

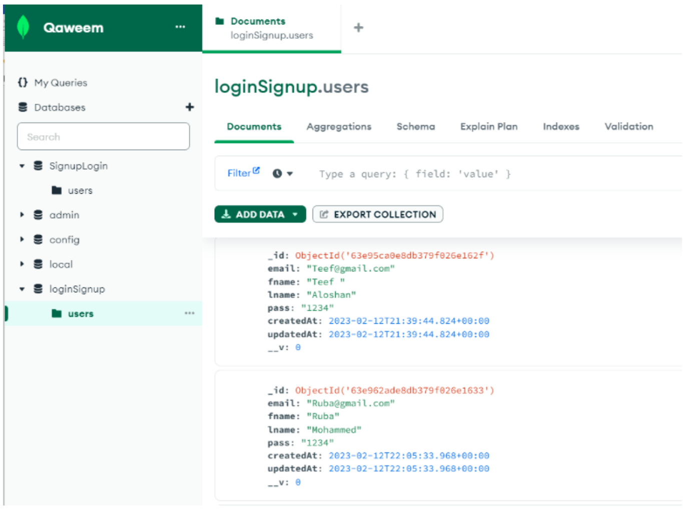
As well while running the program, any sign up or login is recorded and displayed with all the sign up information on terminal like figure3.0.



```
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
(node:13616) [MONGOOSE] DeprecationWarning: Mongoose: the `strictQuery` option will be switched back to `false` by default in Mongoose 7. Use `mongoose.set('strictQuery', fal
se);` if you want to prepare for this change. Or use `mongoose.set('strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
App listening on port http://localhost:5000
mongodb connected.
Teef  Aloshan
Teef@gmail.com 1234 {
  _id: new ObjectId("63e95ca0e8db379f026e162f"),
  email: 'Teef@gmail.com',
  fname: 'Teef ',
  lname: 'Aloshan',
  pass: '1234',
  createdAt: 2023-02-12T21:39:44.824Z,
  updatedAt: 2023-02-12T21:39:44.824Z,
}
Ruba Mohammed
Ruba@gmail.com 1234 {
  _id: new ObjectId("63e962ade8db379f026e1633"),
  email: 'Ruba@gmail.com',
  fname: 'Ruba',
  lname: 'Mohammed',
  pass: '1234',
  createdAt: 2023-02-12T22:05:33.968Z,
  updatedAt: 2023-02-12T22:05:33.968Z,
  __v: 0
```

Figure3.0 (running to collect and store data and display it on virtual studio terminal)

And now I will show where it saved it on MongoDBCompress figure4.0 for the same users as the above figure.



```
_id: ObjectId('63e95ca0e8db379f026e162f')
email: "Teef@gmail.com"
fname: "Teef "
lname: "Aloshan"
pass: "1234"
createdAt: 2023-02-12T21:39:44.824+00:00
updatedAt: 2023-02-12T21:39:44.824+00:00
__v: 0
```

```
_id: ObjectId('63e962ade8db379f026e1633')
email: "Ruba@gmail.com"
fname: "Ruba"
lname: "Mohammed"
pass: "1234"
createdAt: 2023-02-12T22:05:33.968+00:00
updatedAt: 2023-02-12T22:05:33.968+00:00
__v: 0
```

Figure4.0 (saved data on MongoDBCompress)

Problems that we faced and solved:

As we have mentioned before that our first problem was the mongodb version that we downloaded was too low. So, we downloaded a different older version.

Our problem was that the data was not collected and stored in the database that we have created before “SignupLogin” even though the mongodb was connected as shown on the terminal of virtual studio figure 2.0. So, after that after a long time of research we noticed that For signup and login a user collection is needed. For signup we created a new user (email, password, fname, and lname) on the other hand for login we only need with email and password and you can see this on app.js (app.post(‘/register..’), and (app.post(‘/login..’)). After login, we sent the token to the frontend and this is used for login status. If isSuccess is true(login), you can go and browse the website, but if its false you would have to login. As well we created a loginTime which is a duration of how long the user can stay logged intel it automatically logs the user out.

Our Website works, and its full set.

References

1. https://www.getbootstrap.com
2. https://www.w3schools.com/
3. https://youtu.be/znqUwx0b0HI
4. https://youtu.be/exb2ab72Xhs
5. https://www.codinglabweb.com/2023/01/card-slider-html-css.html